

CLAIM AMENDMENT

Please amend the claims as follows:

1. (Canceled)
2. (Canceled)
3. (Canceled)
4. (Canceled)
5. (Canceled)
6. (Canceled)
7. (Canceled)
8. (currently amended) A method for using a first computer system to remotely monitor and interact with the operation of a second computer system through a graphical user interface of said second computer system, comprising the steps of:

receiving a pixel image of said second computer system graphical user interface at

5 said first computer system;

searching said pixel image of said second computer system graphical user interface for

a first graphical element which may be found at one of a plurality of potential
locations an indeterminate location contained within and comprising less than

said pixel image through an automated execution of said first computer system
10 commands;

generating a user peripheral input device input action within said second computer

system graphical user interface as interpreted by a second computer peripheral

input device controller channel by automatically creating and passing a signal

through an i/o communications channel from said first computer system to

15 said second computer system graphical user interface responsive to said

receiving step and results of said searching step steps;

monitoring said pixel image of said second computer system graphical user interface

automatically from said first computer system for an expected second

graphical element contained within and comprising less than said pixel image

20 within a predetermined time interval; and

signaling a failure at said first computer system if said predetermined time interval

elapses without detecting said expected second graphical element.

9. (currently amended) The method of claim 8 further comprising the steps of:

transferring said user input action to a script stored on said first computer system;
re-executing said steps of receiving, generating, monitoring and signaling subsequent
to said storing step under control of said stored script.

10. (Previously presented) The method of claim 8 further comprising the steps of:

providing graphical user interface language extensions commands to a scripting
language; and
passing said generated user input action through said graphical user interface
language extensions from a scripting language processor to a language
extensions processor.

11. (currently amended) The method of claim 8 further comprising the steps of:

generating a user input action within said second computer system responsive to said
second graphical element;
monitoring said second computer system graphical user interface automatically from
said first computer system for an expected third graphical element which may
be found at ~~one of a plurality of potential locations~~ an indeterminate location
contained within and comprising less than said pixel image within a
predetermined time interval; and
signaling a failure at said first computer system if said predetermined time interval

elapses without detecting said expected third graphical element.

12. (Previously presented) The method of claim 8 further comprising the steps of:

depicting said second computer system graphical user interface upon a local display
of said first computer system including said first graphical element; and
receiving a local user input action at said first computer system within said local
display;
wherein said generated user input action emulates said local user input action.

13. (currently amended) The method of claim 8 further comprising the steps of:

providing graphical user interface language extensions commands to a scripting
language; **and**
depicting said computer system graphical user interface upon a local display of said
first computer system including said first graphical element;
receiving a local user input action within said local display;
transferring said user input action to a script stored on said first computer system;
passing said generated user input action through said graphical user interface
language extensions from a scripting language processor to a language
extensions processor for reproduction at said second computer system
graphical user interface, wherein said generated user input action emulates

said local user input action; and
re-executing said steps of receiving, searching, generating, monitoring and signaling
subsequent to said storing step under control of said stored script.

14. (currently amended) A method for enabling a local system to automatically remotely operate
a remote computer system through a graphical user interface on said remote computer system
by using local scripts that selectively respond to changes in graphical displays upon said
graphical user interface of said remote computer system, comprising the steps of:

5 displaying a depiction of said remote system graphical user interface display on said
local system;

capturing user input effected in said depiction of said remote system graphical user
interface display;

10 implementing automatically through a local system command language set user input
emulations that are representative of said captured user input when
reproduced at said remote computer system graphical user interface through
a peripheral input device i/o channel;

15 image processing said remote computer system graphical displays automatically using
a local system script that detects ~~to detect~~ a first entity which may be found
~~at one of a plurality of possible locations~~ an indeterminate location contained
within and comprising less than said graphical display upon said graphical user

20

interface of said remote computer system;
controlling a flow of execution of said local system automatically through a scripting
language having scripting commands in combination with said command
language set, said flow which varies responsive to a result of detection of said
first entity during said image processing step; and
communicating said user input emulations between said local system and said remote
computer system graphical user interface through a communication interface
responsive to said flow controlling step.

15. (Previously presented) The method for enabling a local system to remotely operate a remote
computer system through a graphical user interface on said remote computer system of claim

14 further comprising the steps of:

storing said scripting commands into a storing means;
inserting a command from said command language set into said storing means; and
executing said inserted stored command.

16. (currently amended) A method for using a first computer system to remotely monitor and
interact with the operation of a second computer system through a graphical user interface
of said second computer system, comprising the steps of:

receiving a representation of said second computer system graphical user interface at

5 said first computer system;

searching said representation of said second computer system graphical user interface
for a first graphical entity which may be found at ~~one of a plurality of potential~~
~~locations~~ an indeterminate location contained within and comprising less than
said representation of said second computer system graphical user interface
10 through an automated execution of said first computer system commands; and
generating a user peripheral input device input action within said second computer
system graphical user interface ~~as interpreted by a second computer peripheral~~
~~input device controller channel~~ by automatically creating and passing a
command signal through an i/o communications channel from said first
15 computer system to said second computer system graphical user interface
responsive to said receiving and results of said searching steps;

~~monitoring said second computer system graphical user interface automatically from~~
said first computer system for an expected second graphical entity within a
predetermined time interval; and

20 ~~signaling a failure at said first computer system if said predetermined time interval~~
elapses without detecting said expected second graphical entity.

17. (Canceled)

18. (Previously presented) The method of claim 16 further comprising the steps of:
- providing graphical user interface language extensions commands to a scripting language; and
 - passing said generated user input action through said graphical user interface language extensions from a scripting language processor to a language extensions processor.
19. (currently amended) The method of claim ~~16~~ 37 further comprising the steps of:
- generating a user input action within said second computer system responsive to said second graphical entity;
 - monitoring said second computer system graphical user interface for an expected third graphical entity within a predetermined time interval; and
 - signaling a failure at said first computer system if said predetermined time interval elapses without detecting said expected third graphical entity.
20. (Previously presented) The method of claim 16 further comprising the steps of:
- depicting said second computer system graphical user interface upon a local display of said first computer system including said first graphical entity; and
 - receiving a local user input action at said first computer system within said local display;

wherein said generated user input action emulates said local user input action.

21. (currently amended) The method of claim ~~16~~ 37 further comprising the steps of:

providing graphical user interface language extensions commands to a scripting language; and

depicting said computer system graphical user interface upon a local display of said first computer system including said first graphical entity;

receiving a local user input action within said local display;

transferring said user input action as a command to a script stored on said first computer system;

passing said generated user input action command through said graphical user interface language extensions from said a scripting language processor to a language extensions processor for reproduction at said second computer system graphical user interface, wherein said generated user input action command emulates said local user input action; and

re-executing said steps of receiving, generating, monitoring and signaling subsequent to said storing step under control of said stored script.

22. (Previously presented) The method of claim 8, wherein said step of generating a user peripheral input device input action further comprises locating said user peripheral input

device input action within said first graphical element.

23. (Previously presented) The method of claim 22, wherein said user peripheral input device input action further comprises a click event.
24. (Previously presented) The method of claim 14, wherein said step of communicating between said local system and said remote computer system graphical user interface further comprises locating a user peripheral input device input action at a location relative to said first entity.
25. (Previously presented) The method of claim 24, wherein said user peripheral input device input action further comprises a click event.
26. (Previously presented) The method of claim 16, wherein said step of generating a user peripheral input device input action further comprises locating said user peripheral input device input action within said first graphical entity.
27. (Previously presented) The method of claim 26, wherein said user peripheral input device input action further comprises a click event.
28. (currently amended) A method for enabling a first computing machine to remotely operate

a second computing machine through a graphical user interface on said second computing machine by using computing scripts that selectively respond to changes within graphical displays upon said graphical user interface of said second computing machine, comprising the

5 steps of:

displaying a depiction of said second computing machine graphical user interface display on a graphical user interface of said first computing machine;

capturing user input effected in said depiction of said second computing machine graphical user interface display;

10 image processing said second computing machine graphical displays using a first computing machine script that searches for and detects ~~to detect~~ presence of a first entity which may be found at ~~one of a plurality of potential locations~~ an indeterminate location contained within and comprising less than said graphical display upon said graphical user interface of said second computing machine;

15

controlling a flow of execution of said first computing machine through a scripting language having scripting commands in combination with said a command language set, said flow which varies responsive to a result of detection of said first entity during said image processing step; and

20 implementing user input ~~emulations representative of~~ commands at said second computing machine graphical user interface emulating said captured user input

~~reproduced at said second computing machine graphical user interface through a peripheral input device i/o channel responsive to said first entity presence; and by~~

25 communicating between said first computing machine and said second computing machine graphical user interface through a communication interface responsive to said flow controlling step, a first user input command implemented responsive to a detection of said first entity during said image processing step and a second user input command different from said first user
30 input command implemented responsive to a non-detection of said first entity during said image processing step.

29. (currently amended) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said step of implementing user input ~~emulations~~ commands further comprises locating user input ~~emulations~~ commands at a location determined relative to said first entity.

30. (currently amended) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 29 wherein said step of implementing user input ~~emulations~~ commands

further comprises locating user input ~~emulations~~ commands directly upon said first entity.

31. (Previously presented) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises an icon.
32. (Previously presented) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises a graphical control.
33. (Previously presented) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises a prompt.
34. (Previously presented) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said first entity further comprises a command button.
35. (Previously presented) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second

computing machine of claim 28 wherein said first entity further comprises a message box.

36. (Previously presented) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 28 wherein said second computing machine further comprises a virtual network computing component.

37. (New) The method of claim 16, further comprising the steps of:

monitoring said second computer system graphical user interface automatically from
said first computer system for an expected second graphical entity within a
predetermined time interval; and
signaling a failure at said first computer system if said predetermined time interval
elapses without detecting said expected second graphical entity.

38. (New) A method for using a first computer system to remotely monitor and interact with the operation of a second computer system through a graphical user interface of said second computer system, comprising the steps of:

receiving a pixel image of said second computer system graphical user interface at
said first computer system;
searching said pixel image of said second computer system graphical user interface for

a first graphical element which may be found at one of a plurality of potential and indeterminate locations contained within and comprising less than said pixel image through an automated execution of said first computer system commands;

generating a first user peripheral input device input action within said second computer system graphical user interface by automatically creating and passing a signal through an i/o communications channel from said first computer system to said second computer system graphical user interface responsive to said receiving step and a not-found result of said searching step; and

generating a second user peripheral input device input action within said second computer system graphical user interface different from said first user peripheral input device input action by automatically creating and passing a signal through an i/o communications channel from said first computer system to said second computer system graphical user interface responsive to said receiving step and a found result of said searching step.

39. (New) The method for using a first computer system to remotely monitor and interact with the operation of a second computer system through a graphical user interface of said second computer system of claim 38, further comprising the steps of:

monitoring said pixel image of said second computer system graphical user interface automatically from said first computer system for an expected second graphical element contained within and comprising less than said pixel image within a predetermined time interval; and signaling a failure at said first computer system if said predetermined time interval elapses without detecting said expected second graphical element.

40. (New) A method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine by using computing scripts that selectively respond to changes within graphical displays upon said graphical user interface of said second computing machine, comprising the steps of:

5 displaying a depiction of said second computing machine graphical user interface display on a graphical user interface of said first computing machine;
image processing said second computing machine graphical displays using a first computing machine search command set that searches for and detects presence of a first entity which may be found at one of a plurality of potential
10 and indeterminate locations contained within and comprising less than said graphical display upon said graphical user interface of said second computing machine;
controlling a flow of execution of said first computing machine through a scripting

15 language having scripting commands in combination with a command language set, said flow which varies responsive to a result of detection of said first entity during said image processing step; and

communicating between said first computing machine and said second computing machine graphical user interface through a communication interface responsive to said flow controlling step, a first user input command

20 implemented at said second computing machine responsive to a detection of said first entity during said image processing step and a second user input command different from said first user input command implemented at said second computing machine responsive to a non-detection of said first entity during said image processing step.

41. (New) The method for enabling a first computing machine to remotely operate a second computing machine through a graphical user interface on said second computing machine of claim 40, wherein said first and second user input commands are implemented at said second computing machine graphical user interface by transmitting signals from said first computing machine to said second computing machine graphical user interface through a peripheral input device i/o channel.

42. (New) A method for enabling a first computing machine to remotely operate a second

computing machine through a graphical user interface on said second computing machine by using computing scripts that selectively respond to changes within graphical displays upon said graphical user interface of said second computing machine, comprising the steps of:

- 5 displaying a depiction of said second computing machine graphical user interface display on a graphical user interface of said first computing machine;
- capturing user input effected in said depiction of said second computing machine graphical user interface display;
- developing at least one computing script incorporating user input into said first
- 10 computing machine selected from ones of various scripting functions and commands, said at least one computing script further incorporating said captured user input;
- image processing said second computing machine graphical displays using a first
- computing machine search command set that searches for and detects
- 15 presence of a first entity which may be found at one of a plurality of potential and indeterminate locations contained within and comprising less than said graphical display upon said graphical user interface of said second computing machine;
- controlling a flow of execution of said first computing machine through said at least
- 20 one computing script, said flow which varies responsive to a result of detection of said first entity during said image processing step; and

implementing user input commands at said second computing machine graphical user interface emulating said captured user input by communicating between said first computing machine and said second computing machine graphical user interface through a communication interface responsive to said flow controlling step, a first user input command implemented at said second computing machine graphical user interface responsive to a detection of said first entity during said image processing step and a second user input command different from said first user input command implemented at said second computing machine graphical user interface responsive to a non-detection of said first entity during said image processing step.

43. (New) The method of claim 16, further comprising the step of converting said received representation of said second computer system graphical user interface into a variant different from said second computer system graphical user interface prior to said searching step.
44. (New) The method of claim 43, wherein said step of converting further comprises color translations.
45. (New) The method of claim 44, wherein said color translations further comprises mapping and conversion of said second computer system graphical user interface into particular color

sets based upon geometry.

46. (New) The method of claim 16, wherein said step of searching further comprises determining the location of said first graphical entity when said first graphical entity is found within said second computer system graphical user interface.
47. (New) The method of claim 16, wherein said step of searching further comprises searching for additional occurrences of said first graphical entity within said second computer system graphical user interface when said first graphical entity is found within said second computer system graphical user interface.
48. (New) The method of claim 8, further comprising the step of converting said received pixel image of said second computer system graphical user interface into a variant different from said second computer system graphical user interface prior to said searching step.
49. (New) The method of claim 48, wherein said step of converting further comprises color translations.
50. (New) The method of claim 49, wherein said color translations further comprises mapping and conversion of said second computer system graphical user interface into particular color

sets based upon geometry.

51. (New) The method of claim 8, wherein said step of searching further comprises determining the location of said first graphical element when said first graphical element is found within said second computer system graphical user interface.
52. (New) The method of claim 8, wherein said step of searching further comprises searching for additional occurrences of said first graphical element within said second computer system graphical user interface when said first graphical element is found within said second computer system graphical user interface.
53. (New) The method of claim 14, further comprising the step of converting said depiction of said remote system graphical user interface into a variant different from said remote system graphical user interface prior to said searching step.
54. (New) The method of claim 53, wherein said step of converting further comprises color translations.
55. (New) The method of claim 54, wherein said color translations further comprises mapping and conversion of said remote system graphical user interface into particular color sets based

upon geometry.

56. (New) The method of claim 14, wherein said step of image processing further comprises determining the location of said first graphical entity when said first graphical entity is found within said remote system graphical user interface.
57. (New) The method of claim 14, wherein said step of image processing further comprises detecting additional occurrences of said first graphical entity within said remote system graphical user interface when said first graphical entity is found within said remote system graphical user interface.
58. (New) The method of claim 28, further comprising the step of converting said depiction of said second computing machine graphical user interface into a variant different from said second computing machine graphical user interface prior to said searching.
59. (New) The method of claim 58, wherein said step of converting further comprises color translations.
60. (New) The method of claim 59, wherein said color translations further comprises mapping and conversion of said second computing machine graphical user interface into particular

color sets based upon geometry.

61. (New) The method of claim 28, wherein said searching further comprises determining the location of said first entity when said first entity is found within said second computing machine graphical user interface.
62. (New) The method of claim 28, wherein said searching further comprises searching for additional occurrences of said first entity within said second computing machine graphical user interface when said first entity is found within said second computing machine graphical user interface.
63. (New) The method of claim 38, further comprising the step of converting said received representation of said second computer system graphical user interface into a variant different from said second computer system graphical user interface prior to said searching step.
64. (New) The method of claim 63, wherein said step of converting further comprises color translations.
65. (New) The method of claim 64, wherein said color translations further comprises mapping and conversion of said second computer system graphical user interface into particular color

sets based upon geometry.

66. (New) The method of claim 38, wherein said step of searching further comprises determining the location of said first graphical element when said first graphical element is found within said second computer system graphical user interface.
67. (New) The method of claim 38, wherein said step of searching further comprises searching for additional occurrences of said first graphical element within said second computer system graphical user interface when said first graphical element is found within said second computer system graphical user interface.
68. (New) The method of claim 40, further comprising the step of converting said depiction of said second computing machine graphical user interface into a variant different from said second computing machine graphical user interface prior to said searching.
69. (New) The method of claim 68, wherein said step of converting further comprises color translations.
70. (New) The method of claim 69, wherein said color translations further comprises mapping and conversion of said second computing machine graphical user interface into particular

color sets based upon geometry.

71. (New) The method of claim 40, wherein said searching further comprises determining the location of said first entity when said first entity is found within said second computing machine graphical user interface.
72. (New) The method of claim 40, wherein said searching further comprises searching for additional occurrences of said first entity within said second computing machine graphical user interface when said first entity is found within said second computing machine graphical user interface.
73. (New) The method of claim 42, further comprising the step of converting said depiction of said second computing machine graphical user interface into a variant different from said second computing machine graphical user interface prior to said searching step.
74. (New) The method of claim 73, wherein said step of converting further comprises color translations.
75. (New) The method of claim 74, wherein said color translations further comprises mapping and conversion of said second computing machine graphical user interface into particular

color sets based upon geometry.

76. (New) The method of claim 42, wherein said searching further comprises determining the location of said first entity when said first entity is found within said second computing machine graphical user interface.
77. (New) The method of claim 42, wherein said step of searching further comprises searching for additional occurrences of said first entity within said second computing machine graphical user interface when said first entity is found within said second computing machine graphical user interface.